FORMAT2.WRI

This example allows you to view and manipulate the Formula One palette and then save that palette in VB code for later use as a custom palette. With the example you can manipulate the Red, Green, and Blue values of each palette entry interactively or apply math functions to the palette to create smooth or wild gradients.

The Formula One palette was designed with Excel compatibility in mind. The Excel palette has several (8) entries that are duplicates and can be improved upon for several situations. One is color coordinating a First Impression Chart with Formula One patterns. The Palette1 module was extracted from First Impression. Now that live charts can easily be placed on a Workbook, you can format your data to match chart series colors or even data point colors.

How it works:
When the program is started, the default palette is shown by setting a solid pattern in a cell and then entering the color value in that cell. Color values may be viewed as decimal or hex. The colors are placed on the spreadsheet as they appear in the palettes on the Formula One dialogs.

Selecting a cell will place the Red, Green, and Blue values in the associated text boxes. You can change these values with the associated spinners or by typing a new value in the text box and pressing enter. The dropdown list box provides several palettes different pre-defined palettes.

The Generate Palette frame contains four math functions that can be applied to each of the color values. The generate button will apply the selections to the spreadsheet. At bottom right is a "Save As" button that will save the palette as VB code that may be used later to use the palette in another project.

This project also illustrates how handy Formula One can be as a programming aid. For instance, the Palette1 code was constructed by copying the RGB values from First Impression into columns A, B, and C of a Formula One Workbook. In cell D1 goes the formula

```
=".PaletteEntry(" & ROW() & ") = RGB(" & A1 & ", " & B1 & ", " & C1 & ")"
```

That formula was copied down to row 63. A few entries were removed since the First Impression palette has more entries than the Formula One palette. This was simple since the palette entries were automatically fixed up when the row was deleted. Next, cut and paste into the VB code window which again saved much typing. Another way Formula One is used is as a text formatter. When you save a file in this project, the code is written into Formula One between refreshes and then saved as a text file.

Why use this?
Often it is hard to get just the right look for the finished project. You want just the right colors and patterns for your formatting and spend a lot of time getting that look. This doesn't have much to do with the problem solving you are doing but in many cases is as important. Often you will find colors that suit you and want to use them over. This project provides a little different way of creating a palette but more importantly, allows you to save your work and easily reuse it. This allows you to spend more time on the important code - the code that solves the problem. If you build a collection of these reusable modules, you can leave a lot of the UI till the end and then easliy and quickly change it to suit customer desires.